IoT based Real Time Face Recognition Door Lock System using Neural Network

Khaing Zar Zar Myint
Department of Electronic Engineering, Yangon Technological University, Yangon, Myanmar

Abstract: Face recognition is one of the most popular and challenging topics in the fields of pattern recognition and computer vision. In this system, Raspberry Pi is used for face recognition. Histogram of Oriented Gradients HOG, Principal Component Analysis PCA and Back Propagation Neural Network are used for solving face recognition problem. After solving the face recognition problem, Pi sends the results to the Message Broker Cloud MQTT. It is deal for the "Internet of Things IoT" world of connected devices. NodeMCU is send the result messages which expressed the detected faces are known or unknown by the Cloud MQTT. Finally, Node MCU control the motor to lock or unlock the door according to the messages.

Keywords: *Histogram Equalization, Principal Component Analysis and Back Propagation Neural Network, Raspberry Pi, Cloud MQTT, IoT, NodeMCU.*

I. INTRODUCTION

Face recognition systems can be divided into two groups: dynamic (video) and static matching. In this system, dynamic matching can be used when a video sequence is available. The video images tend to be of low quality, the background is cluttered and more than one face present in the picture. There are many sources of variability in the face recognition problem. They are (a) variation in the image plane, (b) pose variation, (c) lighting and texture variation, (d) background variation and (e) shape variation and so on. In this system, histogram equalization, principal component analysis and back propagation network are used to solve this face recognition problems.

II. OUTLINES OF THE RESEARCH

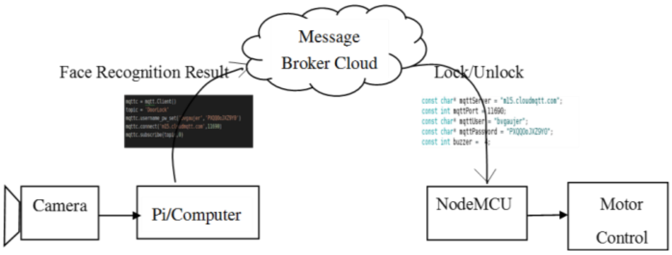

Figure 1: Block Diagram of Face Recognition Door Lock System

Figure 1illustrates the block diagram of face recognition door lock system. Firstly, the Pi camera will capture the face image and will give to the Raspberry Pi/ computer for recognition of the mages. The face detection routine will operate the face region only. The face image is then fed in to the face recognition system which used to recognize by using Neural Network. Then, face recognition result is send to the MQTT cloud configuration to show the determination whether the face is known or unknown. Cloud MQTT is shown in Figure 2. MQTT, machine-to-machine protocol shows the results of the received messages. Message queues provide an asynchronous communications protocol, the sender and receiver of the message do not need to interact with the same message queue at the same time. If the unrecognized face is detected, Cloud MQTT is sent "unknown". And then, the buzzer will alarm 4 seconds and the system will lock door. Figure 3 and 4 illustrate the circuit diagram and hardware of lock/unlock system with NodeMCU. If the recognized face is detected, Cloud MQTT is sent the "KhaingZar" messages which is the name of the known face by the Raspberry Pi. And then, the system will unlock the door. This is shown in Figure 5.

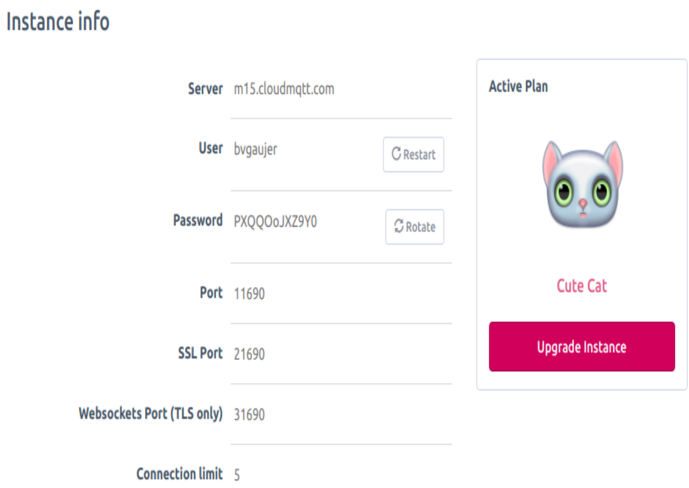

Figure 2: Cloud MQTT

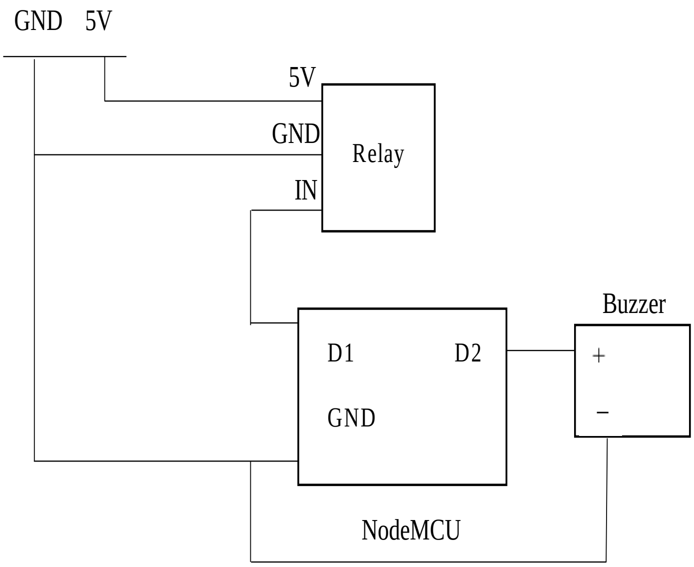

Figure 3: Circuit Diagram of Lock/Unlock System with NodeMCU

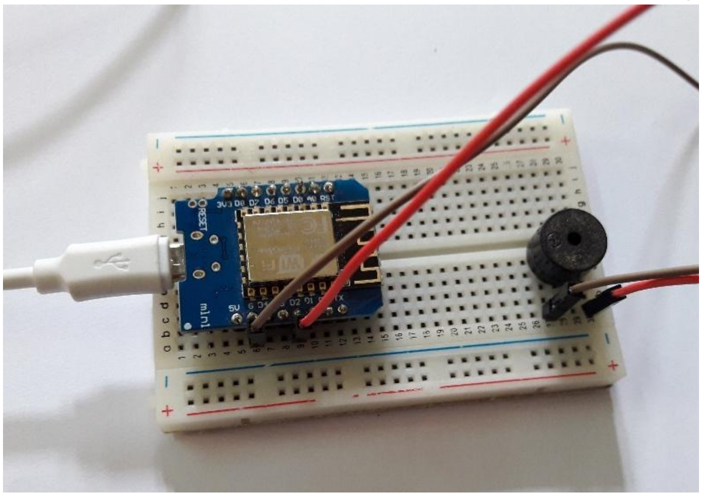

Figure 4: Hardware of Lock/Unlock System with NodeMCU

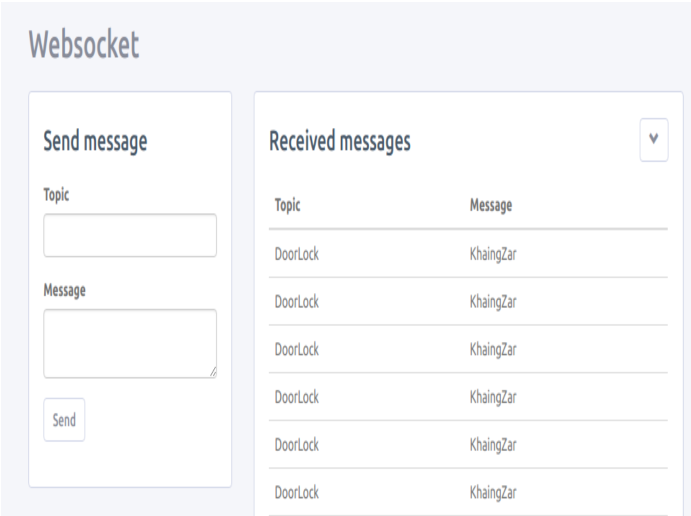

Figure 5: Messages Box of Cloud MQTT

III. METHODOLOGY

Figure 6 shows the flowchart of the real time face recognition system. In this system, there are three basic steps for solving face recognition problem. They are,

- Preprocessing/ Face Detection
- Feature extraction
- Recognition

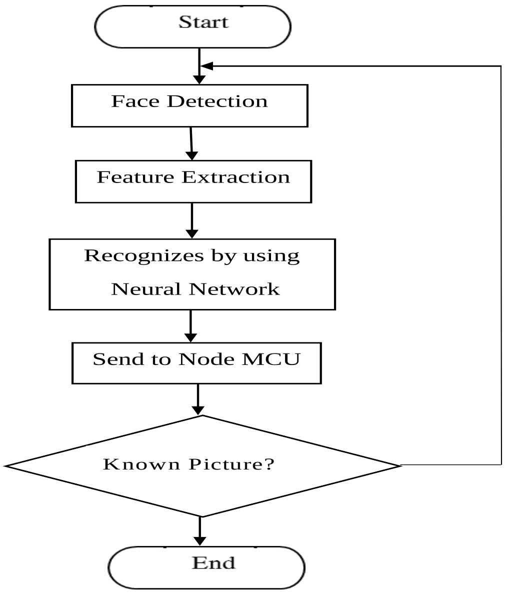

Figure 6: Flowchart of Real Time Face Recognition Door Lock System

Face Detection

The Histogram of Oriented Gradients (HOG) is used for face detection. This is a feature descriptor used in computer vision and image processing for the purpose of object detection. This method is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy. HOG features can be shown not only magnitude but also edge direction or orientation. Therefore, HOG is the most popular feature extraction method of the object detection and recognition fields.[2].

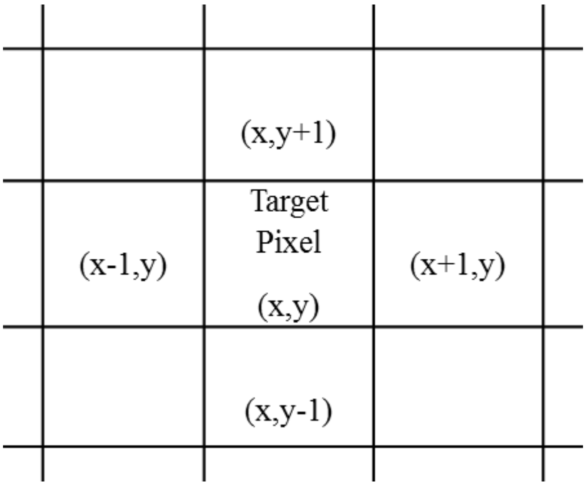

Figure 7: The Table of Target Pixel

$$\nabla f = \begin{bmatrix} f(x,y+1)-f(x,y-1) \\ f(x+1,y)-f(x-1,y) \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \text{.....Eq(1)}$$

To calculate the gradient values of each target pixel, by using Figure 7, the left and the right, the top and the bottom pixel values of target pixels values are used, according to equations (1). And then, magnitudes and directions values of Histogram of Oriented Gradients (HOG) can be calculated by using equations (2) and (3).

The magnitude is $\sqrt{(x^2+y^2)}$ .....Eq(2)

The direction is $\tan^{-1}\left(\frac{x}{y}\right)$ ....Eq(3)

Feature Extraction

In this system, Principal Component Analysis (PCA) is used for feature extraction.PCA is a popular primary technique in pattern recognition and it is used abundantly in all forms of analysis - from neuroscience to computer graphics. PCA can be done by eigen value decomposition of a data covariance (or correlation) matrix or singular value decomposition of a data matrix, usually after a normalization step of the initial data. Equation 4 shows the example of the variance-covariance matrix.[1][6].

$$C = \begin{pmatrix} \mathrm{cov}(x,x) & \mathrm{cov}(x,y) & \cdots & \mathrm{cov}(x,z) \\ \mathrm{cov}(y,x) & \mathrm{cov}(y,y) & \cdots & \mathrm{cov}(y,z) \\ \vdots & \vdots & \vdots & \vdots \\ \mathrm{cov}(z,x) & \mathrm{cov}(z,y) & \cdots & \mathrm{cov}(z,z) \end{pmatrix} \text{..Eq(4)}$$

First, the mean values of each column are calculated by using equation 5.

$$MeanX = \frac{1}{n} \sum_{i=1}^{n} X_i \quad .....Eq(5)$$

$$x_1 = \text{cov}(x,x) - MeanX$$
$$x_2 = \text{cov}(y,x) - MeanX \quad ......Eq(6)$$
$$x_3 = \text{cov}(z,x) - MeanX$$

By that way, $y_1,y_2,....,y_n$ and $z_1,z_2,.....,z_n$ can be calculated. Equation 7 is deviation of respective means.

$$C = \begin{pmatrix} x_1 & y_1 & ...... & z_1 \\ x_2 & y_2 & .... & z_2 \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ x_n & y_n & .... & z_n \end{pmatrix} \quad ......Eq(7)$$

And then, the matrix C must transpose.

$$C^* = \begin{pmatrix} x_1 & x_2 & ...... & x_n \\ y_1 & y_2 & ...... & y_n \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ z_1 & z_2 & ...... & z_n \end{pmatrix} \quad .....Eq(8)$$

The variance-covariance matrix = $\frac{1}{N-1} * C^* * C$ .....Eq(9)

Recognition

In this system, back propagation neural network is used for image recognition. A back propagation neural network is a multi-layer, feed-forward neural network consisting of an input layer, a hidden layer and an output layer. Figure 8 shows the architecture of a back propagation neural network of this system [3][6]. Firstly, initialize the weights and learning rate.

$$O = \frac{1}{1 + \exp[-(\sum x_i w_i - t)]} \quad ......Eq(10)$$

O = output of the threshold element computed using the sigmoidal function
x_i = inputs to the threshold element (i = 1,2,3,……,n)
w_i = weights attached to the inputs
t = threshold for the element. (Assume t = 0)

The output of all hidden layers and final output are calculated by using equation 10. And then, the error E is computed from these output values as

$$E = f(x)_{actual} - f(x)_{output} \quad .....Eq(11)$$

Try to distribute this error to the elements in the hidden layers using a technique called back-propagation. The error of the hidden layers is computed as follows. Let E_j be the error associated with the j^{th} element. Let w_{nj} be the weight associated with the line from element n to element j and let I be the input to unit n. the error for element n is computed as [5]

$$E_n = O_n(1 - O_n) \sum_j w_{nj} E_j \quad ......Eq(12)$$

The associated weights may be updated as

$$w^i_{jk}(new) = w^i_{jk}(old) + \alpha E^{i+1}_k x_{jk} \quad ......Eq(13)$$

α = learning constant
E = associated error measure
x_i = input to the element

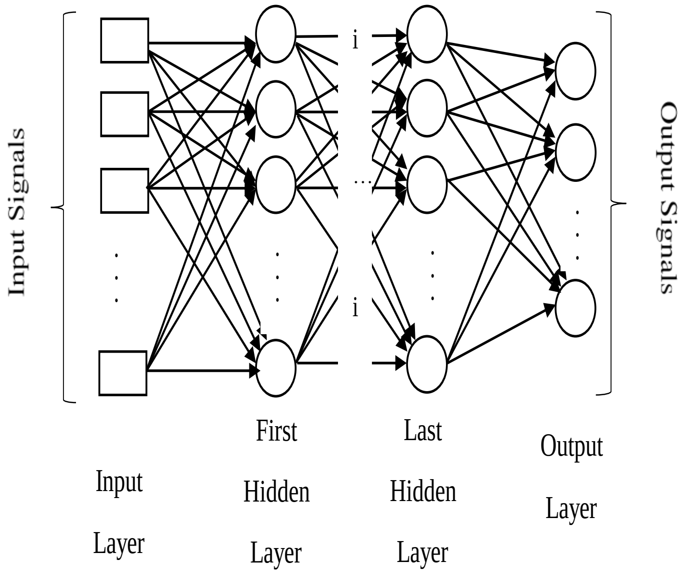

Figure 8: Architecture of a Back Propagation Neural Network

IV. TEST AND RESULTS

Figure 9 (a), (b), (c) and (d) show the detecting images. The detected person wore big eye contacts in Figure 9 (a) and 9 (d). The Cloud MQTT showed the results of these images “KhaingZar”, that is the name of the recognized face. In Figure 9 (b), the person wore spectacles. The received messages showed the name of the detected person. Figure 9 (c) is the natural face of the detected person and its result. The command of these all Figures results are “Door Unlock” which is showed in Figure 10. If the received messages are “unknown”, the buzzer is HIGH level and result command is “Door Lock”. Otherwise, the received message is “KhaingZar”, the door is HIGH level and result command is “Door Unlock”. This is shown in Figure 11.

(a) (b)

(c) (d)

Figure 9: Detected Faces and Results

```
Message arrived in topic: DoorLock
Message:KhaingZar
******Door UnLock ***

-----------------------
Message arrived in topic: DoorLock
Message:KhaingZar
******Door UnLock ***

-----------------------
Message arrived in topic: DoorLock
Message:KhaingZar
******Door UnLock ***

-----------------------
Message arrived in topic: DoorLock
Message:KhaingZar
******Door UnLock ***
```

Figure 10: Received Messages Program for Detected Faces

```
if(msg=="Unknown")
{
  digitalWrite(buzzer, HIGH);
  delay(2000);
  digitalWrite(buzzer, LOW);
  Serial.println("******Door Lock ***");
}
else{
  digitalWrite(door, HIGH);
  delay(10000);
  digitalWrite(door, LOW);
  Serial.println("******Door UnLock ***"}
}
```

Figure 11: Program for Door Lock or Unlock

CONCLUSION

A face preprocessing approach is histogram equalization to improve contrast and compensates for differences in camera input gains. In this paper, Face recognition using Histogram of Oriented Gradients has been shown to be accurate and fast. Multi-neural networks must be trained to deal with all remaining variation (rotation, scale and deformation). When BPNN technique is combined with PCA, non-linear face images can be recognized easily. This method has the acceptance ratio is more than 90 % and execution time of only few seconds. The message broker cloud MQTT operates as a real time communication medium for these IoT based application. MQTT is so lightweight it can be very easily used to transmit the IoT sensor, NodeMCU.

References

[1] Abhishek Singh, Saurabh Kumar, “Face Recognition Using PCA and Eigen Face Approach”, Department of Computer Science and Engineering National Institute of Technology Rourkela, Rourkela-769008, India.

[2] Khamis A. Zidan, “Face Recognition using Neural Network”, https://www.researchgate.net/ Publication/320895494, October 2003.

[3] Shahrin Azuan Nazer, Nazaruddin Omar,Marzuki Khalid, “Face recognition System Using Artificial Neural Networks Approach”, IEEE - ICSCN 2007, MIT Campus Anma University, Chennai, India, Feb, 22 - 24, 2007. Pp 420 – 425 https://www.researchgate.net / publication/ 224698290. March 2007.

[4] Steve Lawrence, Andrew D Back, “Face Recognition: A Convolutional Neural Network Approach”, IEEE Transcations on Neural Networks, Special Issue on Neural Networks and Pattern Recognition, Volume 8, Number 1, pp. 98-113, 1997. https://www.researchgate.net /publication / 3302251. February 1997.

[5] Timothy J. ros, “Fuzzy Logic with Engineering Applications” Third Edition, ISBN: 978-0-470-74376-8, Printed in Singapore by Fabulous Printers Pte Ltd.

[6] P. Latha, Dr. L. Ganesan & Dr. S. Annadurai, “Face Recognition using Neural Networks”, Signal Processing: An International Journal (SPIJ) Volume (3): Issue (5).